Appendix

More Analysis

In this section, we will provide more statistical analysis and show some visualization results to demonstrate the effectiveness of the proposed method.

Selected Scales Distribution. We counted the distributions of the input scales selected by DRS under the entire DLD framework, as illustrated in Table 1. As we can see, the model dynamically chooses different input scales for different images. When γ=0.1, the model is more inclined to accuracy, and more selections are concentrated on the scales of 0.6, 0.7, or 0.8. With the γ increases, the model will select more low-res scales. The scales distributions are also slightly different for different datasets. For example, IC15 chooses more large scales than IC13 since it contains a larger proportion of small text.

Table 1. The distributions of the selected scales using different γ in DLD.

Scales	$\gamma = 0.1$			$\gamma = 0.3$			$\gamma = 0.5$		
	IC13	IC15	TT	IC13	IC15	TT	IC13	IC15	TT
0.3	0.4%	0.2%	0%	4.3%	1.6%	1.6%	17.5%	11.4%	15.0%
0.4	2.6%	0.6%	6.7%	15.9%	14.6%	11.3%	36.0%	35.2%	35.3%
0.5	7.7%	6.2%	7.3%	46.5%	44.2%	47.7%	42.3%	44.4%	42.7%
0.6	24.1%	21.0%	26.3%	21.1%	29.2%	28.7%	3.8%	8.6%	7.0%
0.7	34.8%	39.3%	38.3%	12.0%	10.0%	10.7%	0.4%	0.4%	0%
0.8	30.4%	32.7%	27.4%	0.4%	0.4%	0%	0%	0%	0%

Visualization of the DRS. Dynamic Resolution Selector (DRS) aims to assign different images with different input scales dynamically. Under the candidate scales set $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, we randomly select some of the images from three benchmarks in different scales and visualize them in Figure 1. On the overall trend, we can see that DRS is optimized to choose smaller input scales for images containing large text and bigger input scales for images with small text. Here, the term 'large' or 'small' reflects the relative ratio of the text to the whole image instead of the instance's absolute resolution. The scale that benefits the overall performance would be selected for images containing both large and small text, such as the sample in row-3 and column-1.

Visualization of the SKD. The proposed Sequential Knowledge Distillation (SKD) strategy could help low-resolution images obtain similar performance to those in high-resolution. In Figure 2, we demonstrate some end-to-end results before and after adopting the SKD strategy under the 1/2 resolution inputs (S-384 for IC13, S-640 for IC15 S-448 for TT), respectively. We can see that with the proposed SKD, the model can correctly recognize some of the confused text through context semantic information.

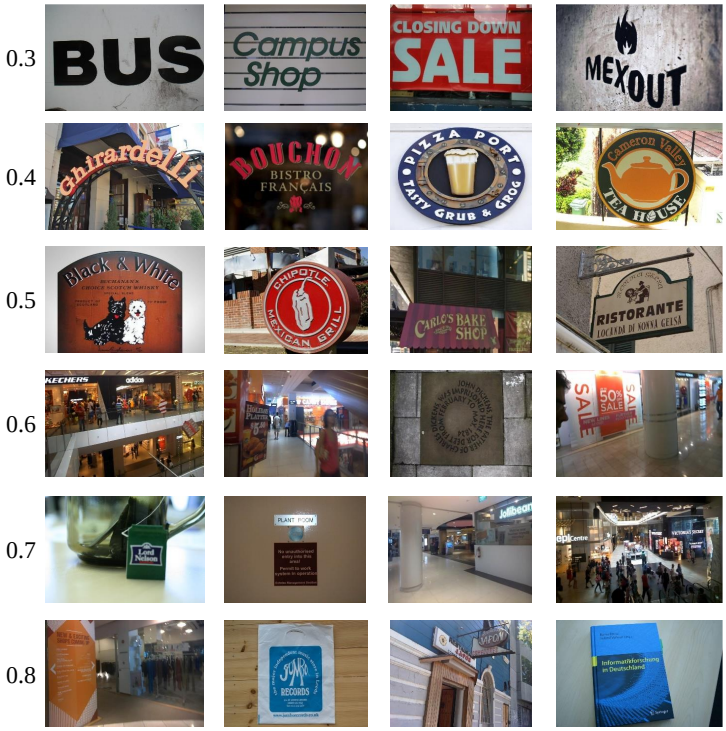

Fig. 1. The selection results of *DLD-DRS-only* for different scales in IC13, IC15, TT.

Visualization of the DLD. Combining with SKD, the model makes the smaller scales have more chances of selecting the DRS module. As for those low-res inputs, DRS can help them select different feasible scales in a cost-efficient way. We illustrate some of the comparisons of End-to-End results between the model of *DLD-SKD-only* and the entire DLD framework in Figure 3. The model of *DLD-SKD-only* inferences images in the 1/2 scales. We can see that although some of the blurred text has been correctly recognized, some text still failed to predict since the infeasible fixed scale, such as the first group of images as shown. Under the entire DLD framework, images can be resized to a suitable scale, achieving better performance with minimal computational cost changes. For those easily recognized images such as the fourth group images in Figure 3, the DLD allows them to choose a lower input scale without performance reduction.

We select some of the samples that are corrected recognized in high-res but failed in low-res by *Vanilla Multi-Scale* model, as visualized in Figure 4. We also provide the compared results predicted by our proposed DLD model. As we can see, under DLD framework, the model has dynamically selected different scales and mostly correctly recognized these instances in the low resolution.

Fig. 2. The visualization of end-to-end results in IC13, IC15, and TT under 1/2 resolution inputs. The first row shows the results without using SKD (*Vanilla Multi-Scale*) and the second row adopts SKD (*DLD-SKD-only*). Text in red are incorrectly recognized.

Feature Distribution of SKD. The target of the SKD is to narrow the feature distribution difference between high resolution and low-resolution inputs. To demonstrate the effectiveness, we visualize the feature distribution in two network positions obtained by Principal Component Analysis (PCA), *i.e.,* (1) *RoI Feature*: the feature map after RoI-Masking operation, and (2) *Contexture Feature*: the feature map after Bi-LSTM module. We compare the feature distribution between high-res and low-res inputs for different models, and the visualization results on three datasets are shown in Figure 5 and Figure 6. As we can see, without using the SKD strategy (*Vanilla Multi-Scale*), the feature differences between high-res and low-res inputs are relatively large. When the model integrates with the SKD (*DLD-SKD-only*), the feature distributions are narrowed. This demonstrates that, although the Multi-Scale training strategy improves the models' robustness on different input scales, the model actually assigns different scales with different feature distributions and results in performance discrepancies when input scales change. The SKD strategy essentially aligns the feature distribution differences and makes the model more robust in low-resolution.

Table 2 records the quantitative differences calculated by L2 distance. Compared with *Vanilla Multi-Scale* training, distillation training can reduce the L2 distance of *RoI Feature* distribution by 0.13 on IC13, 0.11 on IC15 and 0.13 on Total-Text, respectively. As for the *Contextual Feature*, the distances can be decreased by 0.08, 0.08, 0.10, respectively. Comparing the feature discrepancies

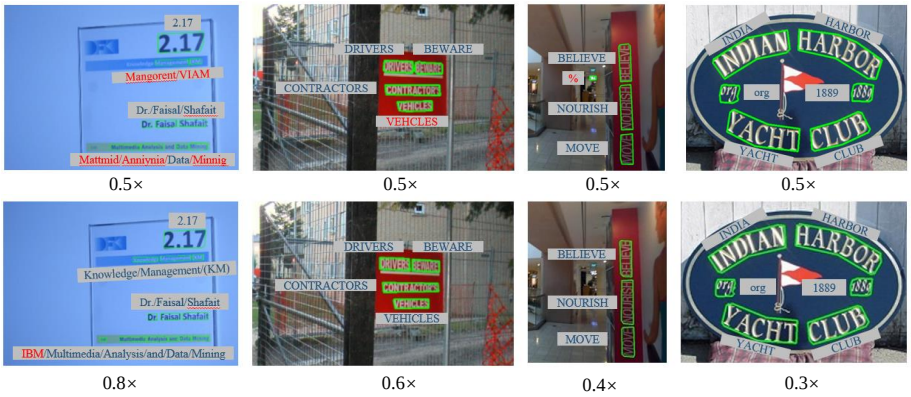

Fig. 3. The visualization of end-to-end results in IC13, IC15, and TT. The first row shows the results under 1/2 input scales in *DLD-SKD-only* and the second row corresponds to the entire *DLD*. The numbers below the images are the down-sampled scales compared to the original high-resolution. Text in red are incorrectly recognized.

MS	H: Produkt	H: TOAST	H: DESSERT	H: BEACH
	L(0.5x): Prodult	L(0.5x): TOKST	L(0.5x): BERGERS	L(0.5x): BEACE
DLD	L(0.4x): Produkt	L(0.6x): TOAST	L(0.8x): DESSERT	L(0.5x): BEACH

MS	H: CHANDRA	H: Family	H: Hougang	H: Ave
	L(0.5x): GLASSI	L(0.5x): Fandy	L(0.5x): Haugang	L(0.5x): Are
DLD	L(0.4x): CHANDRA	L(0.5x): Family	L(0.6x): Haugang	L(0.6x): Ave

Fig. 4. The compared recognition results between *Vanilla Multi-Scale* and *DLD*. Characters in red are incorrectly recognized.

in two positions, the RoI feature's differences between high-res and low-res inputs are much larger than the other. This demonstrates that it is unwise only to adopt distillation in the deepest layer. Distillation on RoI features can help the model recover the information in the early phases and prevent over-fitting.

Discussion

The technique of End-to-End text spotting has been studying for several years since [4] was proposed. Many advances have demonstrated the advantages of End-to-End models compared to the traditional two-staged pipeline, for example, (1) better performance since freeing from error accumulation between two tasks, (2) faster inference and smaller storage requirements by information sharing and jointly optimization (3) lower maintenance cost [1].

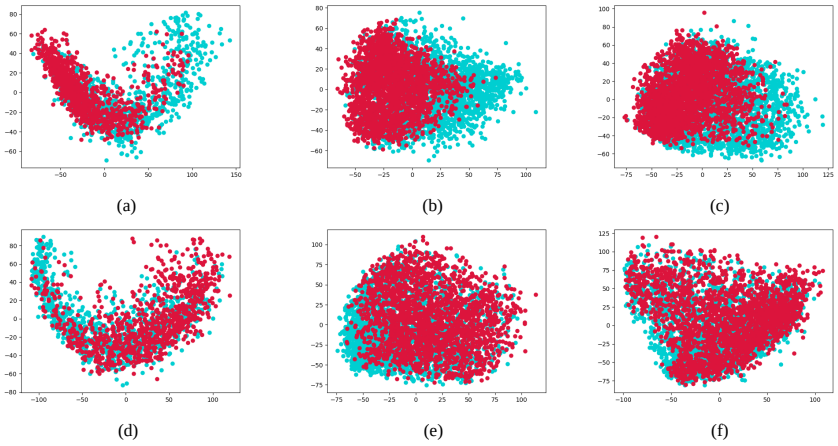

Fig. 5. The PCA visualization of the *RoI Feature* distributions on high-res and low-res inputs. (a) *Vanilla Multi-Scale* on IC13. (b) *Vanilla Multi-Scale* on IC15. (c) *Vanilla Multi-Scale* on TT. (d) *DLD-SKD-only* on IC13. (e) *DLD-SKD-only* on IC15. (f) *DLD-SKD-only* on TT. Blue points denote the high resolution distribution and Red points denote the low resolution distribution.

However, most of the current Optical Character Recognition (OCR) engines still solve the real tasks in a two-staged way, such as Tesseract [6], PP-OCR [2], Calamari [7], *etc*. There might exist many reasons, but one of the immediate factors is that an end-to-end model with good performance and robustness is hard to obtain in many real complicated situations. Due to the different characteristics of the sub-tasks, it is not easy to balance two tasks [8]. The text detection task predicts instances' scopes and locations and focuses more on the different scales of coarse-grained features like boundaries. The recognition task is a sequential classification problem that usually requires images on a uniform scale, and it is more concentrated on fine-grained features like textures. Therefore, the input resolution becomes a sensitive factor to the second staged task and may cause the overall performance fluctuation.

To alleviate such problems, besides the intuitive way that makes the model be trained on more samples and with abundant means of augmentations, the proposed DLD framework can effectively reduce the resolution sensitivity of the text recognition task from the feature level. In real applications, the DLD can be flexibly used to enhance the robustness of the end-to-end model.

Failure cases. In this section, we demonstrate some failure cases in the testing phase. The failure cases are mainly caused by wrong resolution selection and errors inherent in the detection model, here we only focus on the failure cases caused by the wrong resolution selection to reveal the limitations of our proposed

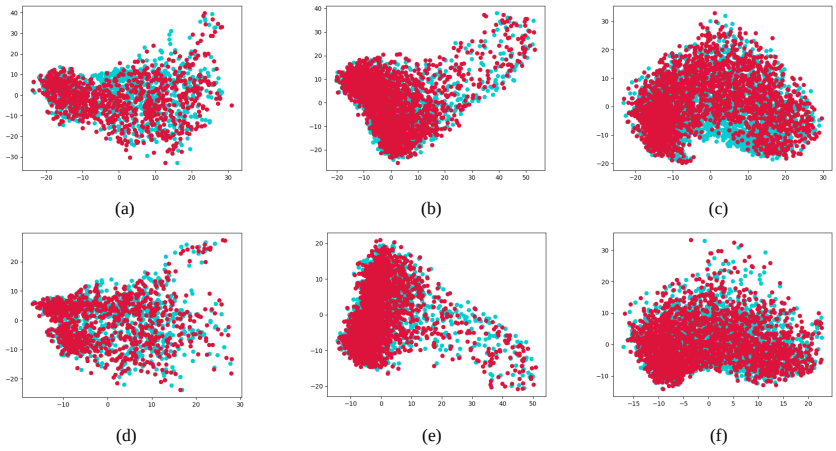

Fig. 6. The PCA visualization of the *Contexture Feature* distributions on high-res and low-res inputs. (a) *Vanilla Multi-Scale* on IC13. (b) *Vanilla Multi-Scale* on IC15. (c) *Vanilla Multi-Scale* on TT. (d) *DLD-SKD-only* on IC13. (e) *DLD-SKD-only* on IC15. (f) *DLD-SKD-only* on TT. Blue points denote the high resolution distribution and Red points denote the low resolution distribution.

framework. As shown in 7, we can see that it is difficult for the resolution selector to choose a suitable resolution when the text size distribution in the images varies greatly. Moreover, there are also a few failure cases assigning large resolutions to images with large text or assigning small resolutions to images with small text, which will brings extra time consumption or makes the text unrecognizable. Nonetheless, our proposed resolution selector works well for most text images and reduces the input resolution without affecting the performance. To demonstrate the effectiveness, as shown in Table 3, we illustrate the distribution of failure cases. We can see from the table that most of the failure cases are caused by errors inherent in the detection model. Moreover, errors due to the introduction of our proposed *DLD* framework only account for 2.51% in IC13, 2.16% in IC15 and 1.94% in TT, respectively. Furthermore, equipped with our proposed sequential distillation, we can effectively correct some of incorrect recognition results.

Limitations. Although the proposed DLD framework effectively optimizes the model's performance of accuracy and computational cost, there are still some problems to be solved in the future.

First, the basic high-resolution and the set of candidate down-sampled scales need to be decided manually, which requires sufficient data analysis and certain experiences of people. Given an empirical setting, although the model can perform better than simply using multi-scale training and fixed-scale testing, but can hardly find out the theoretically best solution. Enlarging the candidate set's

Dataset	Training Method	L2 distance of RoI Feature	L2 distance of Contexture Feature
IC13	Vanilla Multi-Scale	0.43	0.20
	DLD-SKD-only	0.30	0.12
IC15	Vanilla Multi-Scale	0.28	0.18
	DLD-SKD-only	0.17	0.10
TT	Vanilla Multi-Scale	0.34	0.19
	DLD-SKD-only	0.21	0.09

Table 2. L2 distances of feature distribution between high-res and low-res inputs on different benchmarks.

Dataset	*Vanilla Multi-Scale*	*DLD*	ratio (%)
IC13	✔	✔	78.08
	✔	✗	2.51
	✗	✔	3.49
	✗	✗	15.92
IC15	✔	✔	62.93
	✔	✗	2.16
	✗	✔	4.72
	✗	✗	30.19
TT	✔	✔	64.79
	✔	✗	1.94
	✗	✔	4.74
	✗	✗	28.53

Table 3. The distribution of failure cases in IC13, IC15 and TT. 4 means correct text recognition, 7 means incorrect text recognition.

capacity is a way to find more optimal results. However, the training cost will inevitably increase since the selector needs to calculate the forward results of all candidate scales during training. A possible alternative way to achieve dynamic resolution selecting is to predict resolutions in soft labels rather than hard labels, such as quality assessment [3]. However, we still face many challenges since it is hard to define the quality of an image containing different text.

Second, Although the KD strategy can increase the performance of the low-res model, there is always a limit to the competence of the model. When the input image is down-sampled to a small scale that even humans cannot recognize, the KD learning might lead the model to an over-fitting state. Furthermore, the current model only considers the KD problem on the recognition task. However, the detection task is also sensitive to the resolution for the text spotters that adopt the segmentation-based detection branch. Although we have demonstrated the effectiveness to adopt our method on those models, the performances still have large improvement spaces. More experiments should be conducted on the different types of models in the future.

The last problem is a common problem for most current text spotters. The target of text spotting is to obtain the final text sequences prediction. However, the text detection branch is mostly optimized to obtain high-IoU with the detection Ground Truth (GT), which is not always consistent with text recogni-

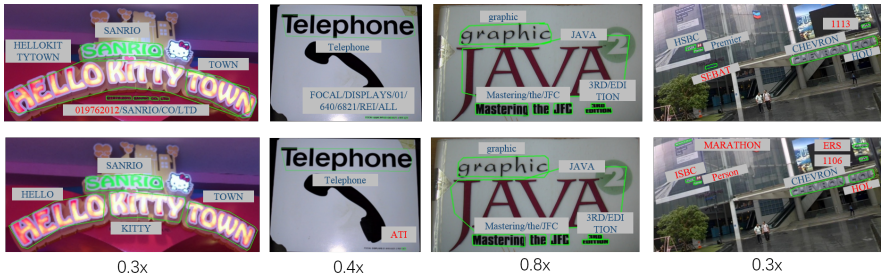

Fig. 7. Visualization of some failure cases caused by wrong resolution selection in IC13, IC15, and TT. The first row shows the results under original input scales with *Vanilla Multi-Scale* and the second row corresponds to the entire *DLD*. The numbers below the images are the down-sampled scales compared to the original high-resolution. Text in red are incorrectly recognized.

tion [5,8]. This problem would introduce the new question about how to balance these two tasks. In our DLD, in addition to the GT-oriented optimization, many other balance parameters still need to be set manually, which might also influence the model's performance to some extent.

References

1. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: A survey. ACM Comput. Surv. **54**(2), 42:1–42:35 (2021) 4
2. Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., Wang, H.: PP-OCR: A practical ultra lightweight OCR system. CoRR **abs/2009.09941** (2020) 5
3. He, L., Gao, F., Hou, W., Hao, L.: Objective image quality assessment: a survey. Int. J. Comput. Math. **91**(11), 2374–2388 (2014) 7
4. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: ICCV. pp. 5248–5256 (2017) 4
5. Qiao, L., Chen, Y., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F.: MANGO: A mask attention guided one-stage scene text spotter. In: AAAI. pp. 2467–2476 (2021) 8
6. Smith, R.: An overview of the tesseract OCR engine. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil. pp. 629–633. IEEE Computer Society (2007) 5
7. Wick, C., Reul, C., Puppe, F.: Calamari - A high-performance tensorflow-based deep learning package for optical character recognition. CoRR **abs/1807.02004** (2018) 5
8. Zhong, H., Tang, J., Wang, W., Yang, Z., Yao, C., Lu, T.: ARTS: eliminating inconsistency between text detection and recognition with auto-rectification text spotter. CoRR **abs/2110.10405** (2021) 5, 8